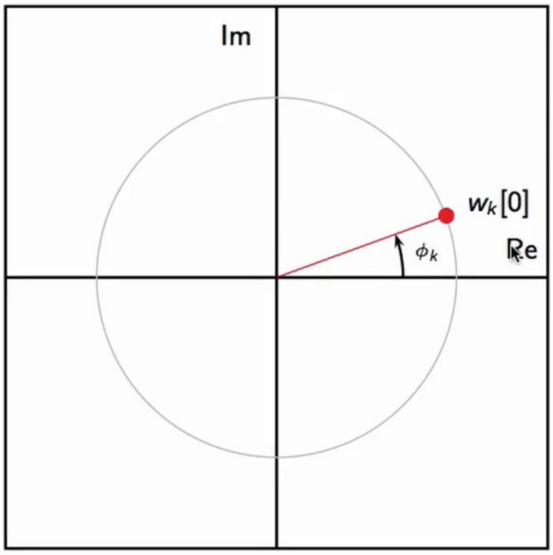
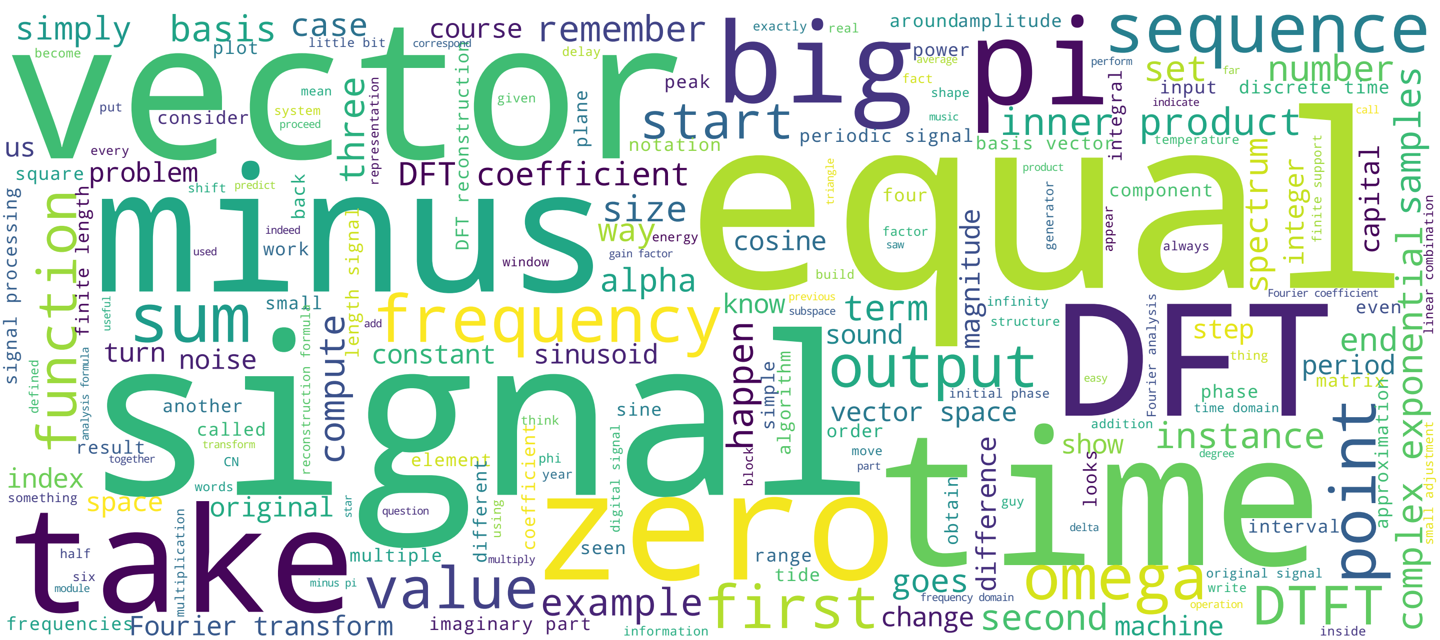
Synthesis: the sinusoidal generator

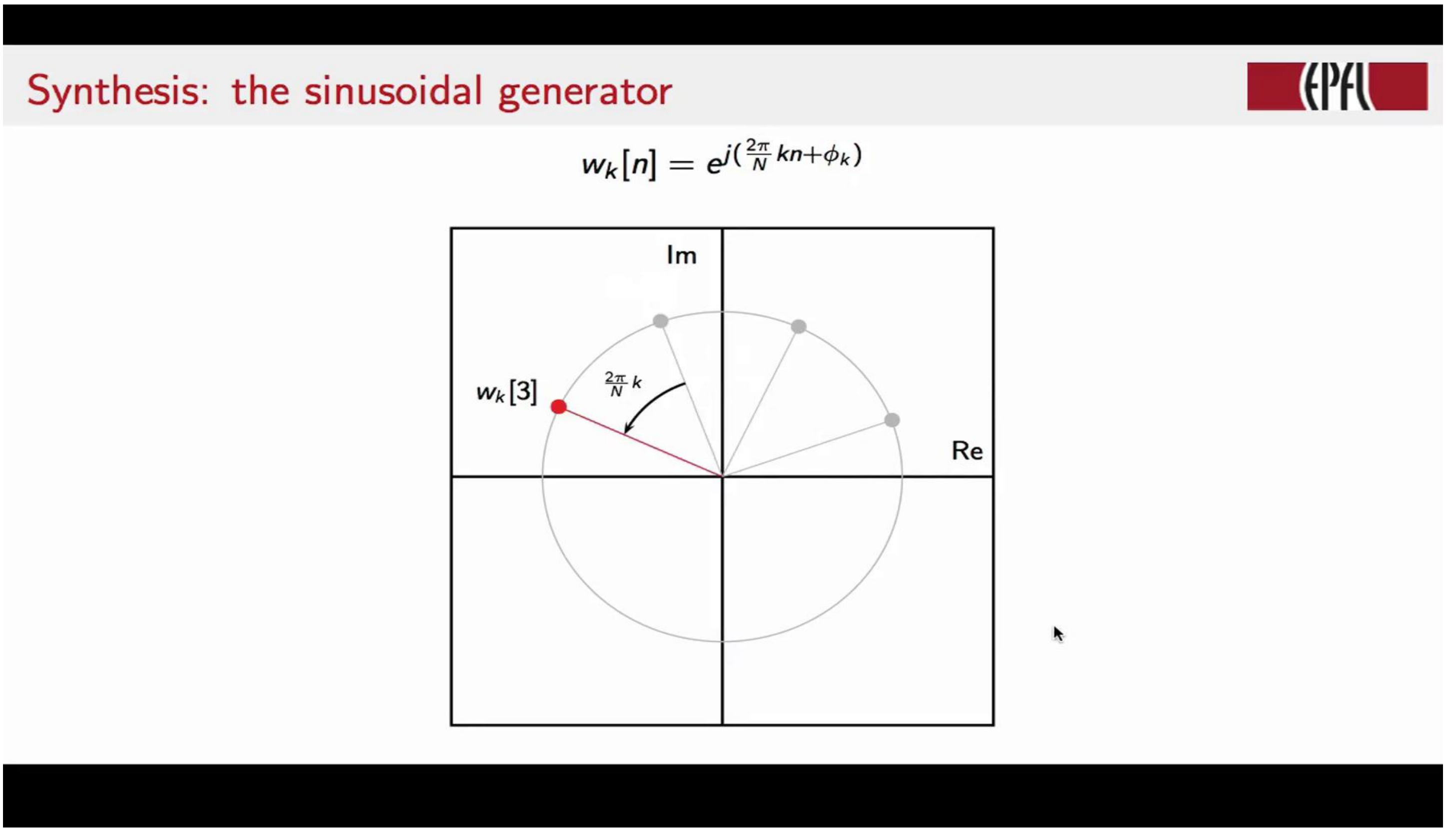
$$w_k[n] = e^{j(\underbrace{\frac{2\pi}{N}}\, kn + \underbrace{\phi_k})}$$





Search MOOC

Video



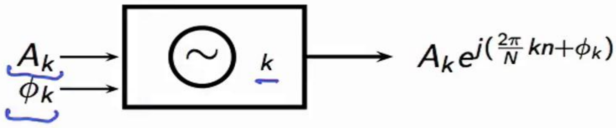
Okay, so far we have looked at DFT as an analysis tool. Let's now spend some time thinking about the synthesis properties of the DFT, how we build the signals starting from DFT coefficients. To better understand how the DFT reconstruction formula works, it's useful to look at the standard sinusoidal generator that works at one of the standard frequencies in Fourier basis. Here you have a complex exponential with a frequency which is a multiple of two pi over big N, and with an initial phase 5k. The way this generator works is by successively generating points on the unit circle starting at the phase 5k and proceeding in increments of two pi over big N.

Notes

Summary

0m 00s

Synthesis: the sinusoidal generator

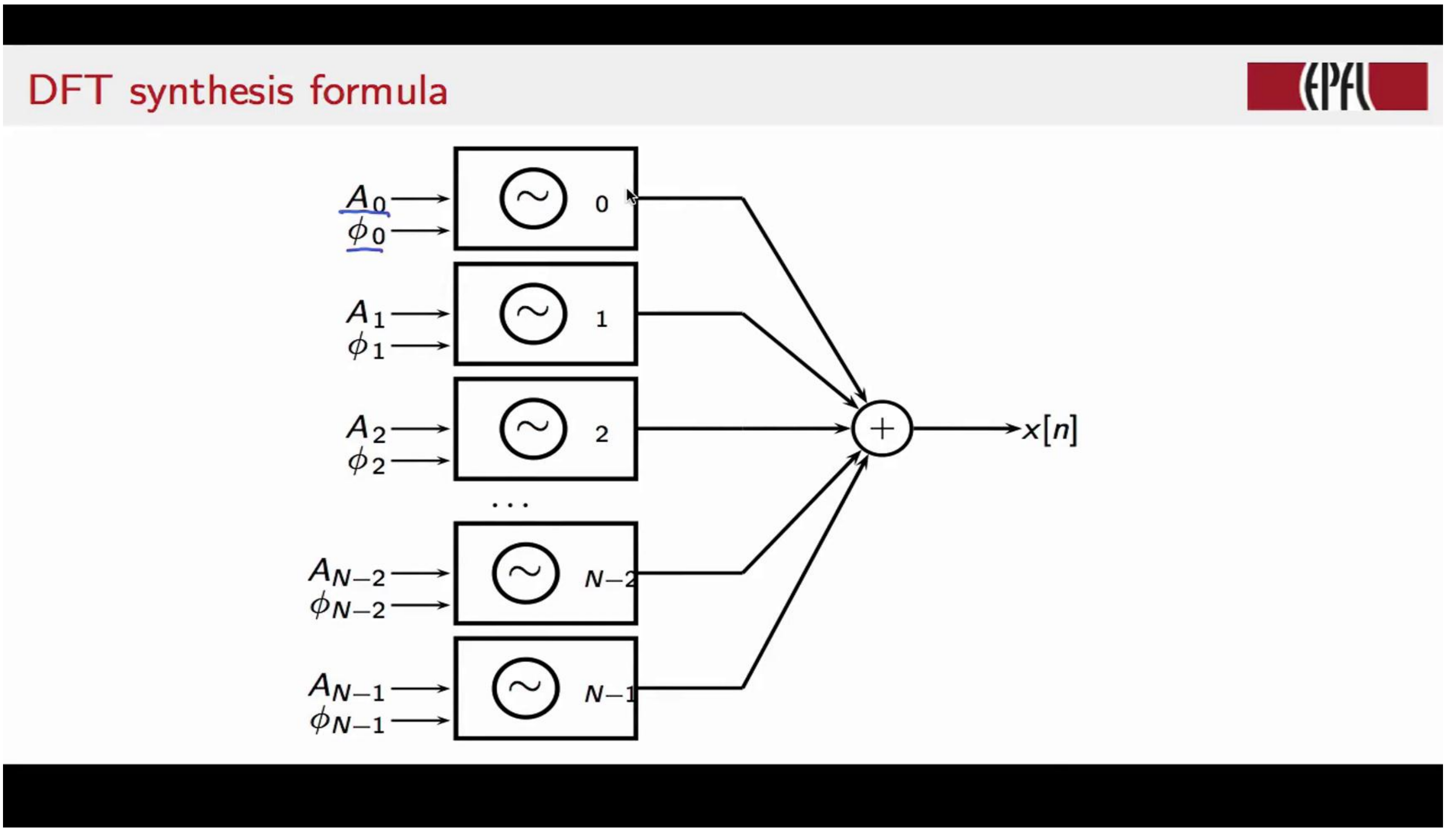


We can draw this concisely with this notation. Where here inside you have the index of the Fourier frequency that the generator will produce. The input parameters for the generators are, of course, the initial phase here and a gain* factor A of K that will lead to A of K times to e to the j to pi over big N, Kn plus phi k.

Notes

Summary

0m 45s



So the DFT reconstruction formula can be drawn graphically like a machine composed of big N sinusoidal generators, each one of which will be initialised with a gain factor and with a phase factor. All their outputs will be sum together and this will give us back the original signal. How do we initialise each one of these blocks?

Notes

Summary

1m 07s

Initializing the machine

$$A_k = |X[k]|/N$$

$$\phi_k = \angle X[k]$$

Well, the amplitude will be the amplitude of the corresponding Fourier analysis coefficient normalised by big N. The phase will be the phase of the corresponding Fourier analysis coefficient.

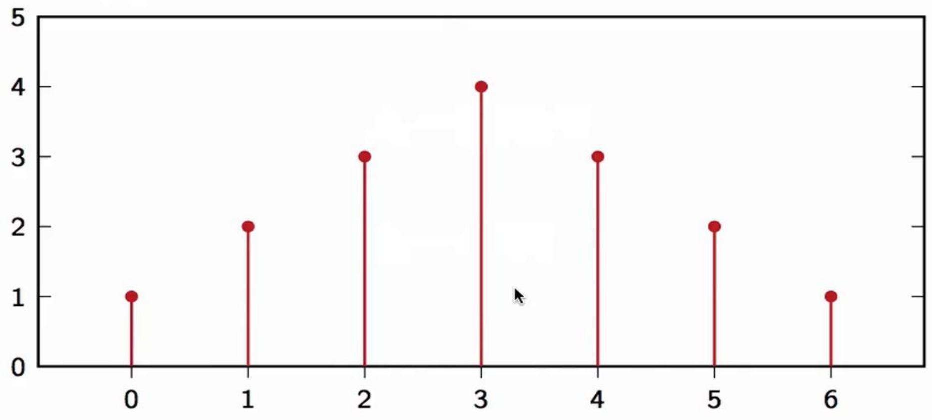
Notes

Summary

1m 29s

Example

$$\mathbf{x} = [1\ 2\ 3\ 4\ 3\ 2\ 1]^T$$



So let's look at an example. Let's take a simple signal, a seven point long signal that looks like a triangle.

Notes

Summary

1m 40s

Example

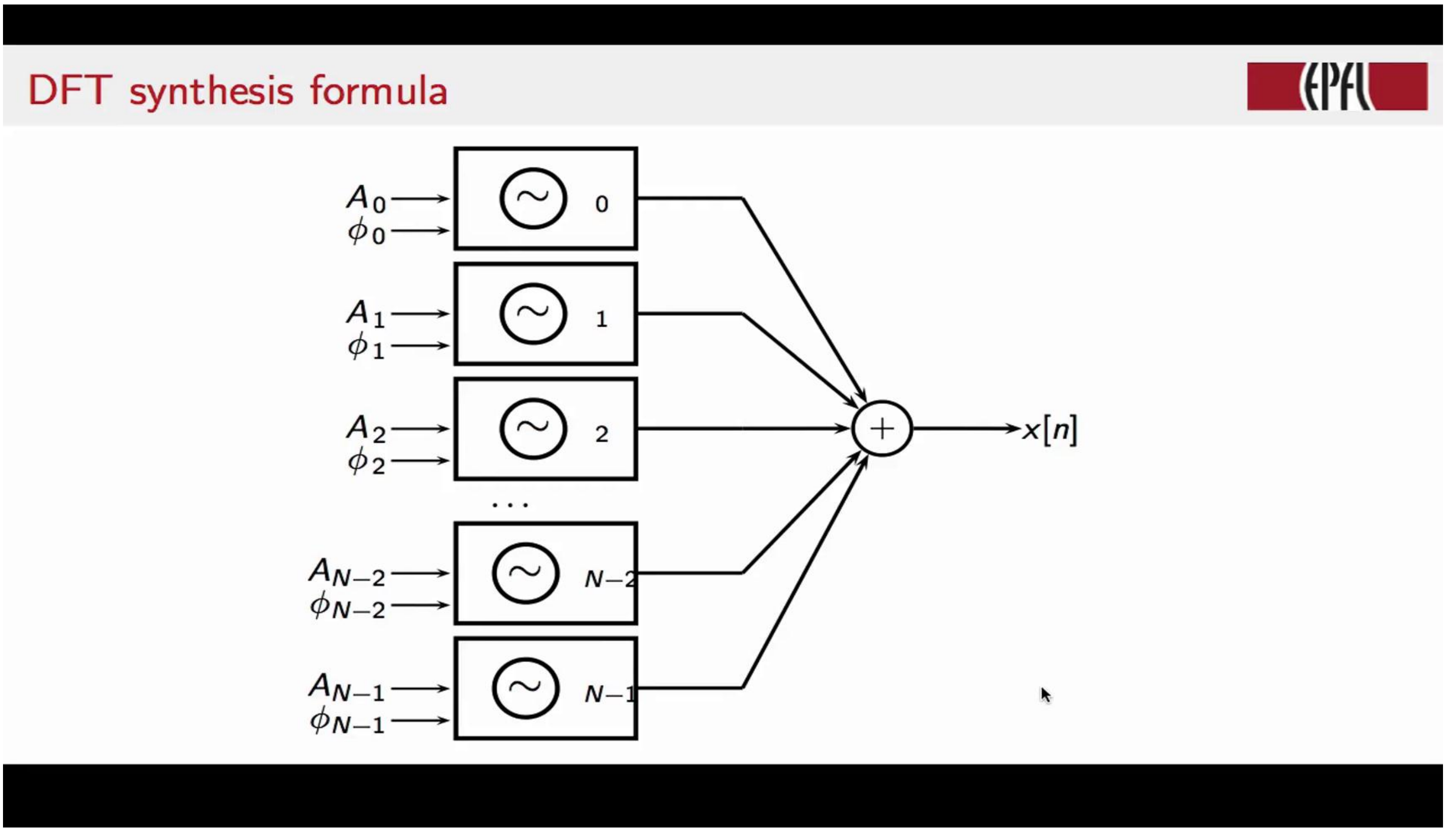
k	A_k	ϕ_k
0	2.2857	0.0000
1	0.7213	-2.6928
2	0.0440	0.8976
3	0.0919	-1.7952
4	0.0919	1.7952
5	0.0440	-0.8976
6	0.7213	2.6928

And we perform a Fourier analysis on this vector and we get seven Fourier coefficients that we list here in terms of amplitude and phase.

Notes

Summary

1m 49s

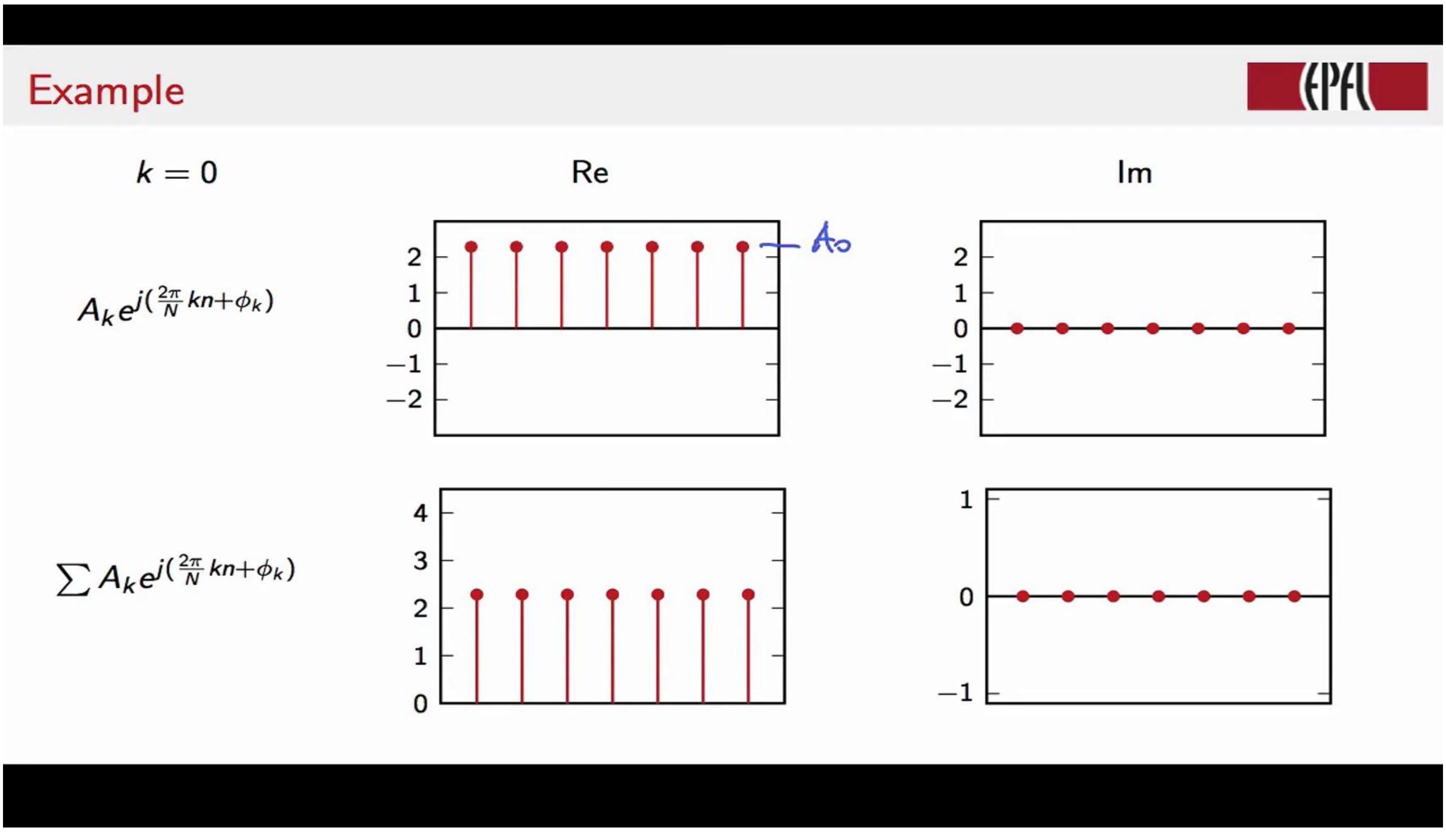


Now, we initialise the seven generators with the A's and the phi's that we found in the analysis part. Then we turn the crank seven times and we will sum the outputs together to obtain the original signal.

Notes

Summary

2m 01s

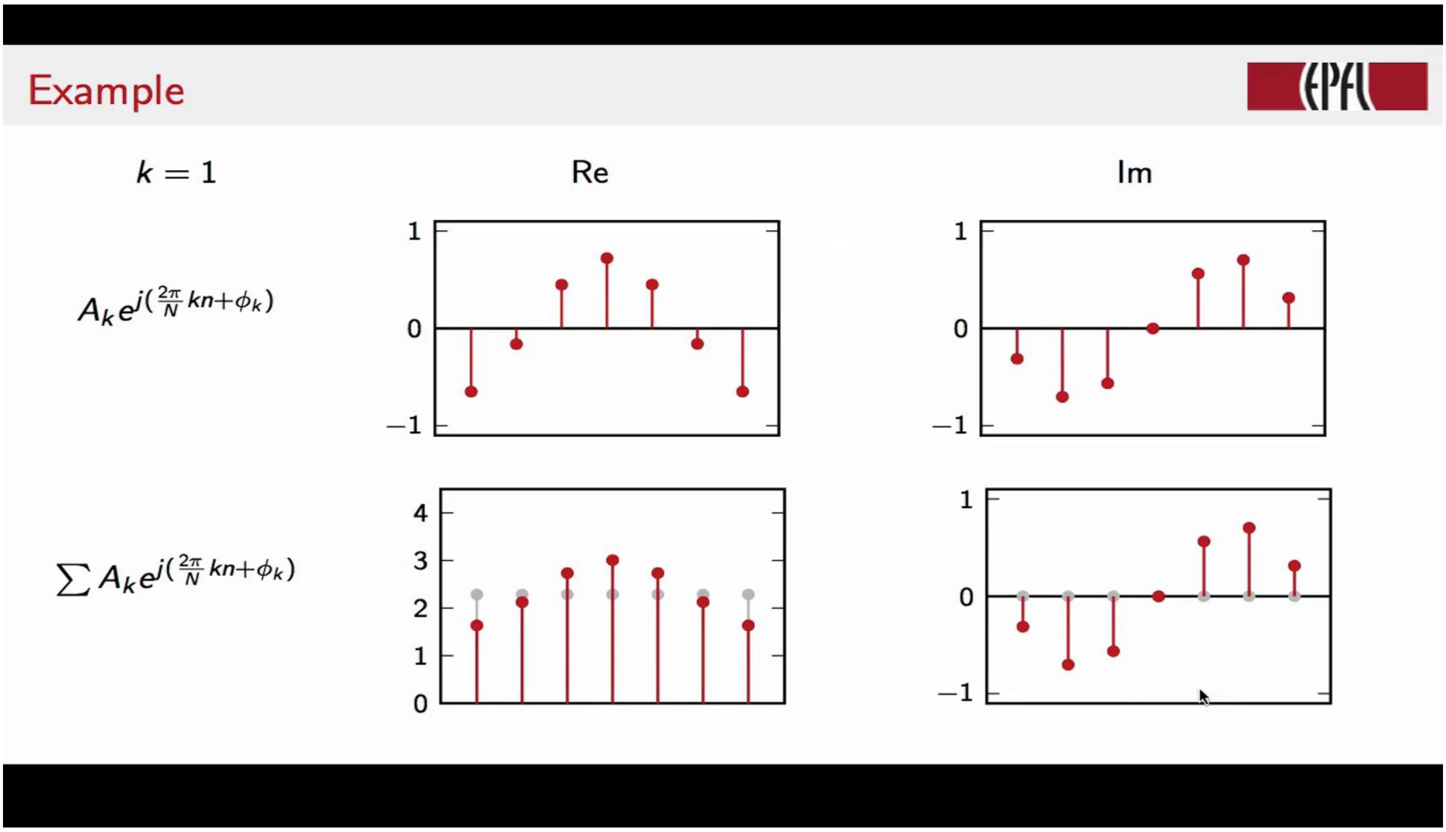


Here is how it works. In the first line, we will show the output of each generator in turn, starting from k equal to zero up to k equal to six. Here on the bottom line, we will show the cumulative sum of the outputs as k proceeds from 0-6. So the first generator is the one for k equal to zero, so the frequency is zero and the output will be simply a constant. As a matter of fact, the value of this constant will be equal to A of zero. The cumulative sum now has only one term, so the output at this point will look like a constant.

Notes

Summary

2m 13s



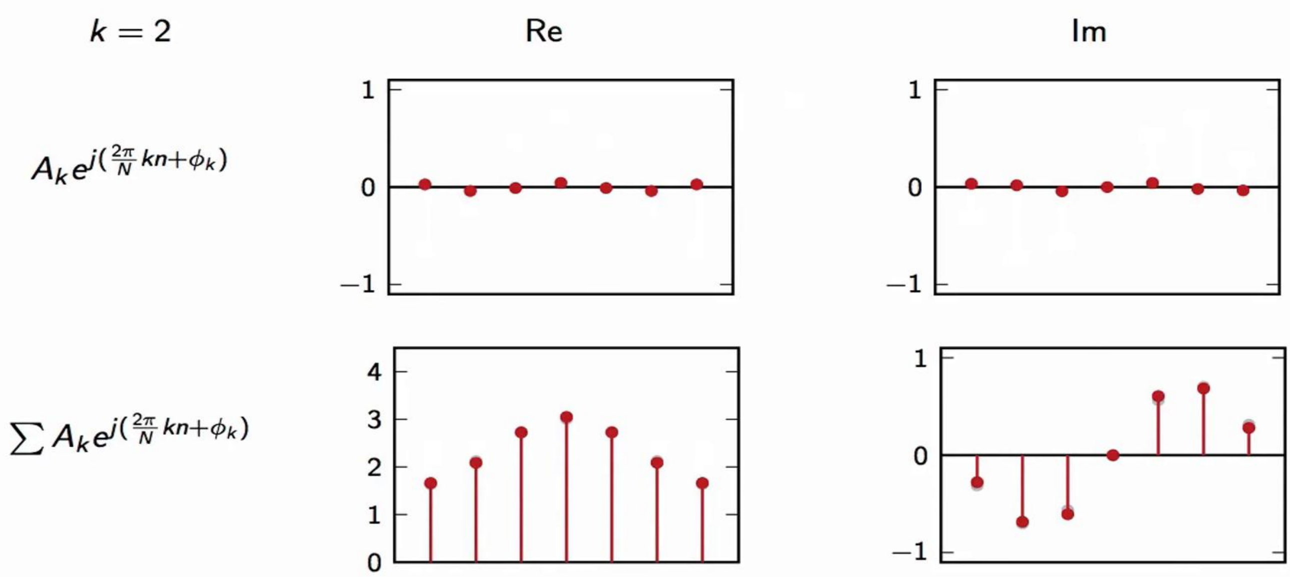
For k equal to one, we have now a sinusoid that will have a certain initial phase and a certain amplitude. And we can see that as we add this to the previous constant, here in gray you see the previous sum, and here the updated sum for k equal to one. The real part of the signal starts to look a little bit like a triangle. We also have a significant, non-negligible imaginary part, which will have to disappear in the end, because remember, we started from a real signal.

Notes

Summary

2m 51s

Example

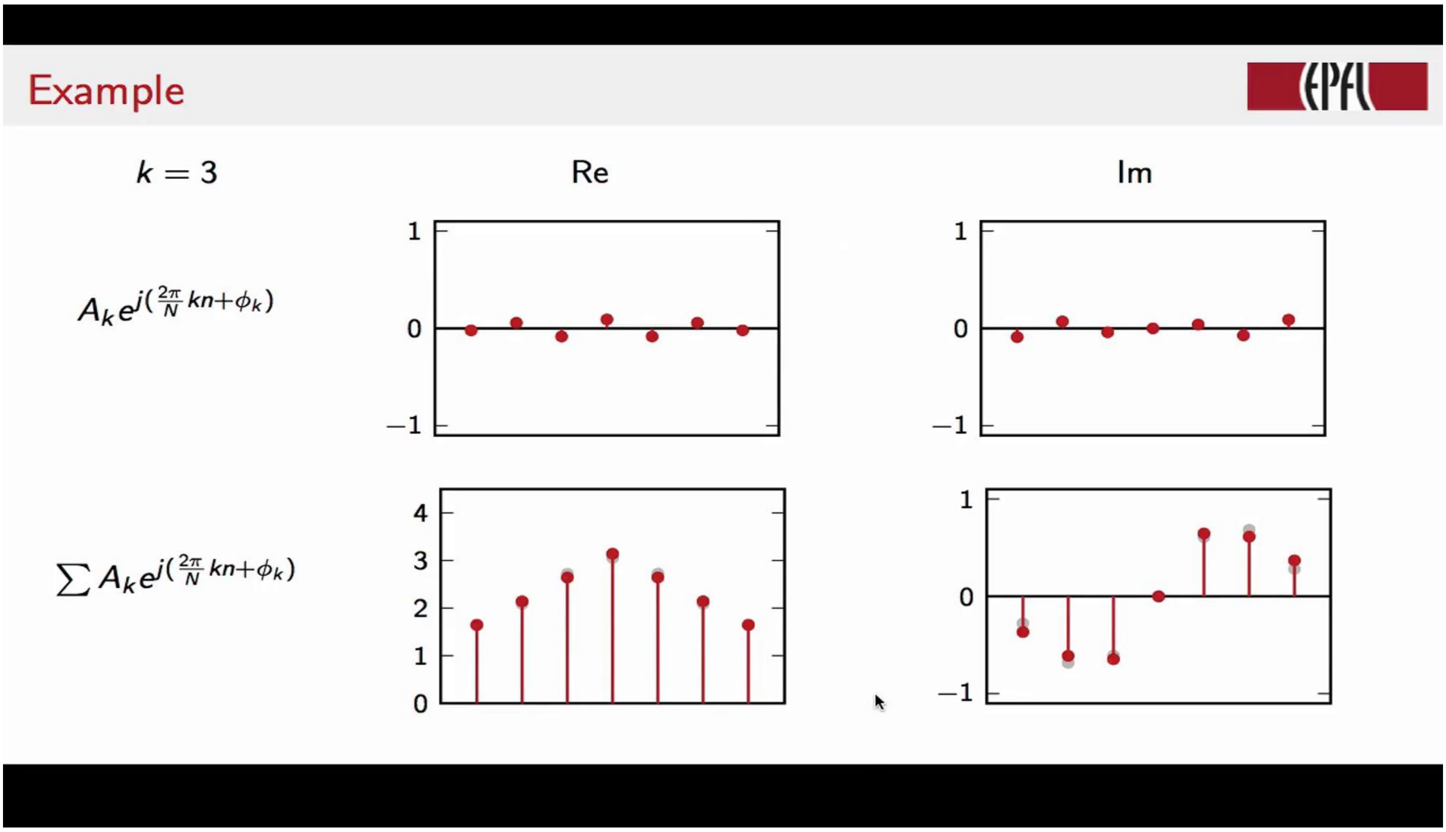


As we proceed for k equal to two, we see that the coefficient amplitude is very small and so this will be just small adjustments in the sum. It's very hard to see the difference between this step and the previous step.

Notes

Summary

3m 19s

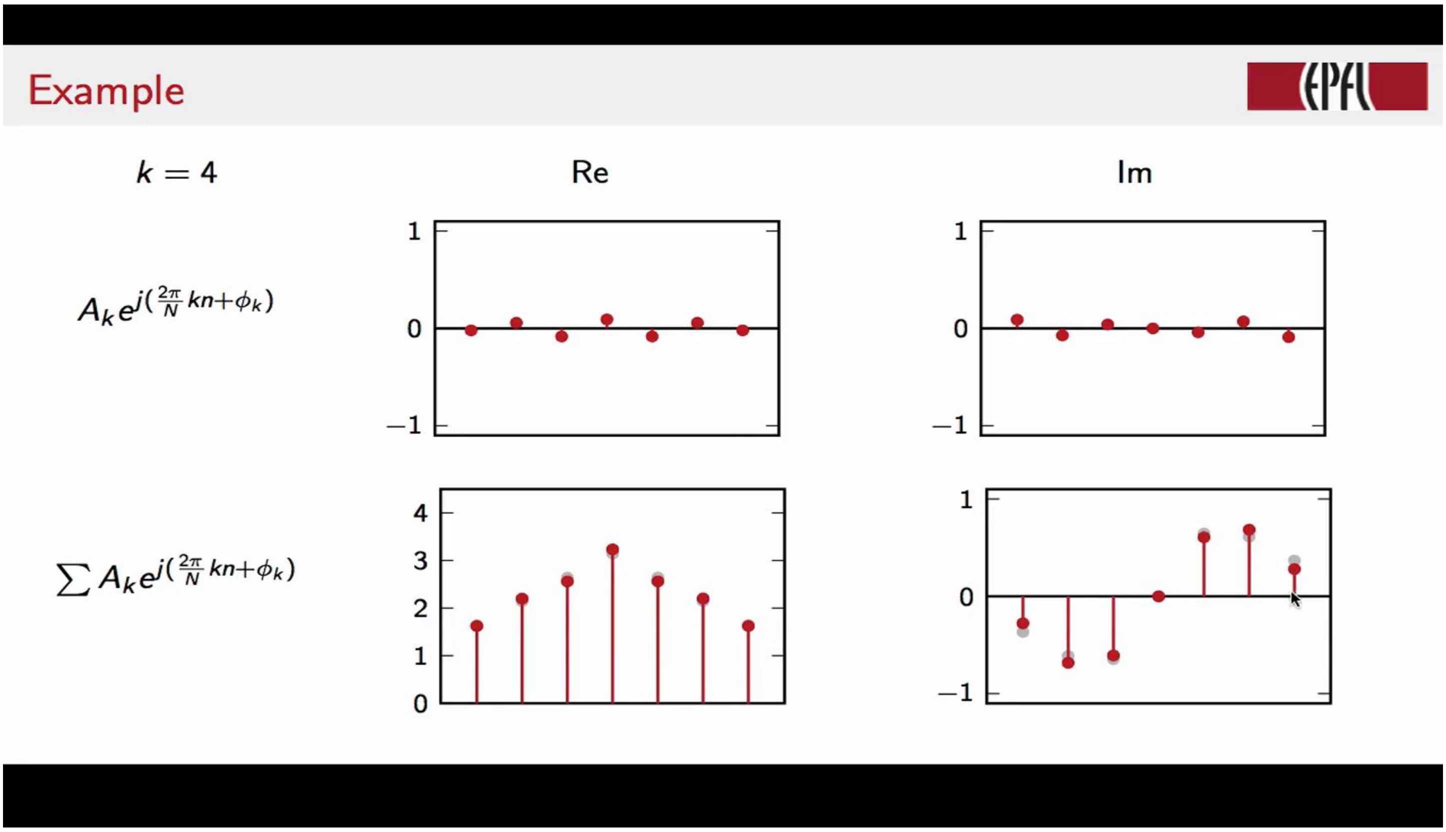


K equal to three is very similar, small adjustments.

Notes

Summary

3m 31s

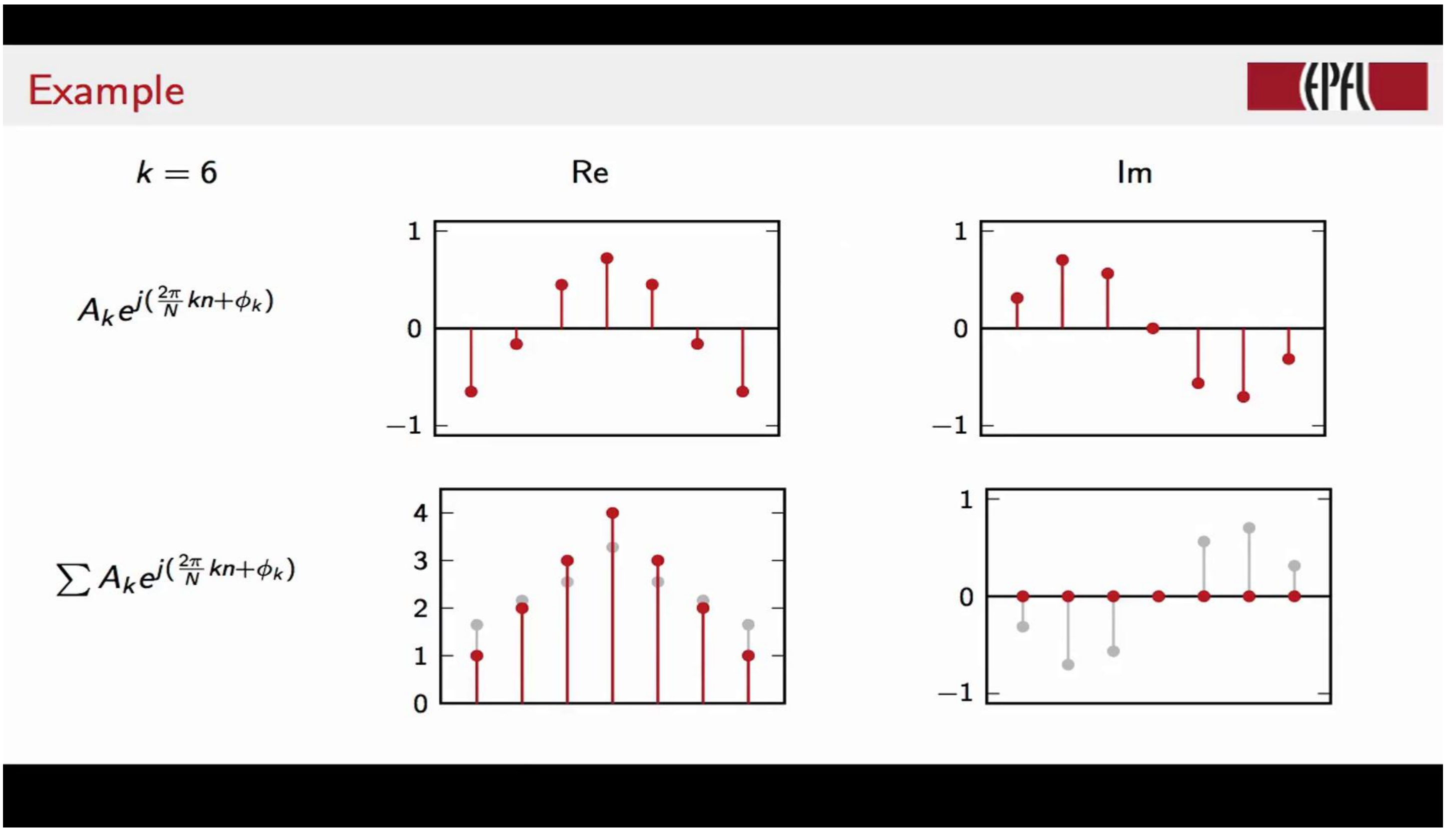


For k equal to four, we're going in the counterclockwise part of the spectrum, so more adjustments still that will start to bring down the imaginary part to zero.

Notes

Summary

3m 34s



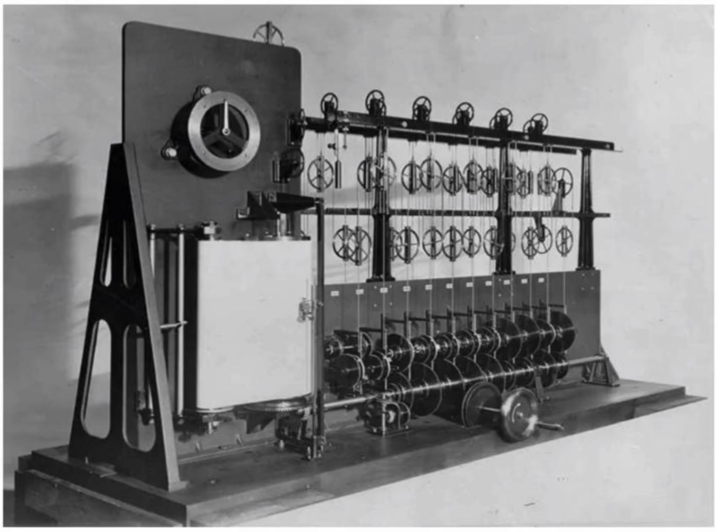
Here you see another small adjustment, and finally, k equal to six, the last coefficient will completely kill the imaginary part and give us back exactly the signal we were starting from. Why would we think about the DFT reconstruction formula as a machine?

Notes

Summary

3m 45s

The machine before DSP



tide-predicting machine (originally invented by Lord Kelvin)

Because in reality, these machines were used in the past, for instance, to predict the tides. The tide is a periodic phenomenon, and once you find the modes, you can actually use the experimental data to predict the evolution of the tide as a superposition of sinusoidal components. This machine that was originally invented by Lord Kelvin, is nothing but a mechanical implementation of the DFT reconstruction machine that we just saw before.

Notes

Summary

4m 00s

Running the machine too long...

$$x[n+N] = x[n]$$

output signal is N-periodic!

Here's a question. What happens if you turn the crank of the machine more than big N times? Well, it turns out that the output will become periodic. In other words, x of small n plus big N will be equal to x of n.

Notes

Summary

4m 27s

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n = 0, 1, \ldots, N-1$$

produces an N-point signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k = 0, 1, \ldots, N-1$$

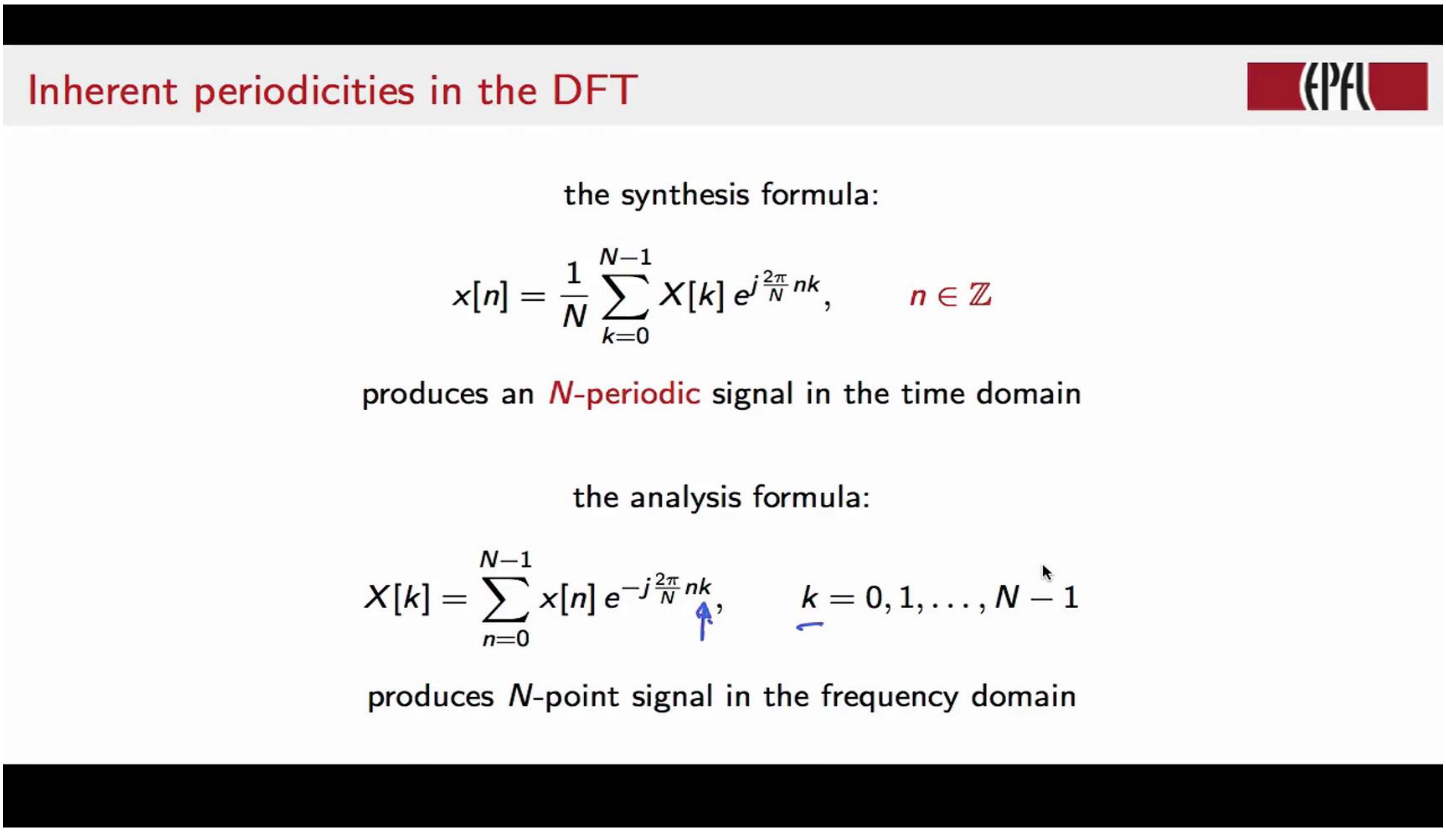
produces N-point signal in the frequency domain

This is actually quite apparent from the structure of the synthesis and analysis formulas of DFT. Let's start from the synthesis formula. In theory, the output index small n should go from zero to big N minus one. But you can see that it only appears here inside this complex exponential. Now, this guy is too pi periodic, so if you push n over big N minus one, this will simply cycle over once again as if n was looping over the zero to n minus one range.

Notes

Summary

4m 39s



In the end, you can actually safely take n to be from the set of integers, and the output will be an N-periodic signal in the time domain. Actually, the same holds for the analysis formula. You can let the index k roam over the entire set of integers. And since it appears only in this complex exponential here, again it will be as if k was looping over the zero to n minus one range.

Notes

Summary

5m 10s

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n \in \mathbb{Z}$$

produces an *N*-periodic signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k \in \mathbb{Z}$$

produces *N*-periodic signal in the frequency domain

What happens really, is that you can consider the sequence of DFT coefficient as an N-periodic signal in the frequency domain.

Notes

Summary

5m 37s